

Network Performance Measurements for NASA's Earth Observation System

Joe Loiacono
Computer Sciences Corporation
NASA Goddard Space Flight Center
Greenbelt, Maryland

Andy Germain*
Swales Aerospace, Inc.
NASA Goddard Space Flight Center
Greenbelt, Maryland

Jeff Smith
NASA Goddard Space Flight Center
Greenbelt, Maryland

Abstract

NASA's Earth Observation System (EOS) Project [1] studies all aspects of planet Earth from space, including climate change, and ocean, ice, land, and vegetation characteristics. It consists of about 20 satellite missions over a period of about a decade. Extensive collaboration is used, both with other U.S. agencies (e.g., National Oceanic and Atmospheric Administration (NOAA), United States Geological Survey (USGS), Department of Defense (DoD), and international agencies (e.g., European Space Agency (ESA), Japan Aerospace Exploration Agency (JAXA)), to improve cost effectiveness and obtain otherwise unavailable data. Scientific researchers are located at research institutions worldwide, primarily government research facilities and research universities.

The EOS project makes extensive use of networks to support data acquisition, data production, and data distribution. Many of these functions impose requirements on the networks, including throughput and availability. In order to verify that these requirements are being met, and be pro-active in recognizing problems, NASA conducts on-going performance measurements. The purpose of this paper is to examine techniques used by NASA to measure the performance of the networks used by EOSDIS (EOS Data and Information System) and to indicate how this performance information is used.

Keywords: network measurement, network performance, data distribution

* This work was partially supported by the National Aeronautics and Space Administration under contract number NAS5-01909 to Swales Aerospace, Inc.

1. Introduction

NASA Earth Science (ES) missions typically involve a satellite in low Earth orbit (LEO) that hosts a small number of remote-sensing instruments, all of which collect data pertaining to a single scientific theme. For example, the Aura mission [2], scheduled for launch in 2004, is an atmospheric research mission that will host four instruments. LEO satellites circle the Earth at about 200 to 500 miles high, completing an orbit in approximately 90 to 100 minutes. Once or twice each orbit, when the satellite is in contact with one of the ground stations, data collected by the instruments is downlinked to the ground. After some initial processing, the data products are sent to one of the Distributed Active Archive Centers (DAACs). These DAACs then perform higher-level processing, to generate data products more useful to researchers. There are seven DAACs, located throughout the U.S., which store data products from ES missions and which support interactive and interoperable retrieval and distribution of these data products. Table 1 presents a list of the DAACs, their locations, and the type of information stored at each one.

The purpose of EOSDIS is to provide scientific and other users access to data from NASA's Earth Science Enterprise. These users are located throughout the world. Several types of networks make up EOSDIS, including NASA Integrated Services Network (NISN) [3], various research networks (such as the Internet2 Abilene network [4]), and international networks to reach international partners. Figures 1 and 2 show the sites serviced by EOSDIS within the U.S., and internationally, respectively. These sites include, of course, NASA centers involved in Earth Science, other Federal agency partners, international agency partners, and the DAACs.

At NASA Goddard Space Flight Center (GSFC) we have developed an EOSDIS network performance measurement system, called "ENSIGHT" (EOS Networks Statistics and Information Gathering HTML-based visualization Tool) [5]. In this paper we address network monitoring of data transfer to and from the DAACs. Transmission of data to support instrument control and satellite downlink from spacecraft are both outside the scope of this paper. Two classes of measurements are taken: passive measurements and active measurements. Passive measurements consist of data about actual "user" flows, collected from operational network elements, generally routers. Passive metrics are not intended to add significant flows to the network, although the results are generally acquired "in-band." Active measurements, on the other hand, do intentionally add traffic to the network, to measure the response. Thus passive measurements generally reflect user operations, while active measurements show a snapshot of the network capabilities available beyond the level of usage at the time.

Table 1. Distributed Active Archive Centers

Name	Location	Type of Data
Physical Oceanography	Jet Propulsion Laboratory, Pasadena, California	Oceanic processes, air-sea interactions
Atmospheric Sciences Data Center	NASA Langley Research Center, Hampton, Virginia	Radiation budget, clouds, aerosols, tropospheric chemistry
Snow and Ice	National Snow and Ice Data Center (NSIDC), Boulder, Colorado	Snow and ice, cryosphere, climate
Land Processes	Earth Resources Observation Systems (EROS) Data Center (EDC), Sioux Falls, South Dakota	Land-related data
Alaska Synthetic Aperture Radar (SAR) Facility (ASF)	University of Alaska, Fairbanks, Alaska	SAR data, sea ice, polar processes, geophysics
Goddard Space Flight Center (GSFC) Earth Sciences (GES)	NASA Goddard Space Flight Center, Greenbelt, Maryland	Upper atmosphere, atmospheric dynamics, global land biosphere, global precipitation, ocean color
Oak Ridge National Laboratory (ORNL) DAAC**	Oak Ridge National Laboratory, Oak Ridge, Tennessee	Biogeochemical and ecological data useful for studying environmental processes

** This DAAC is not serviced by EOSDIS networks.

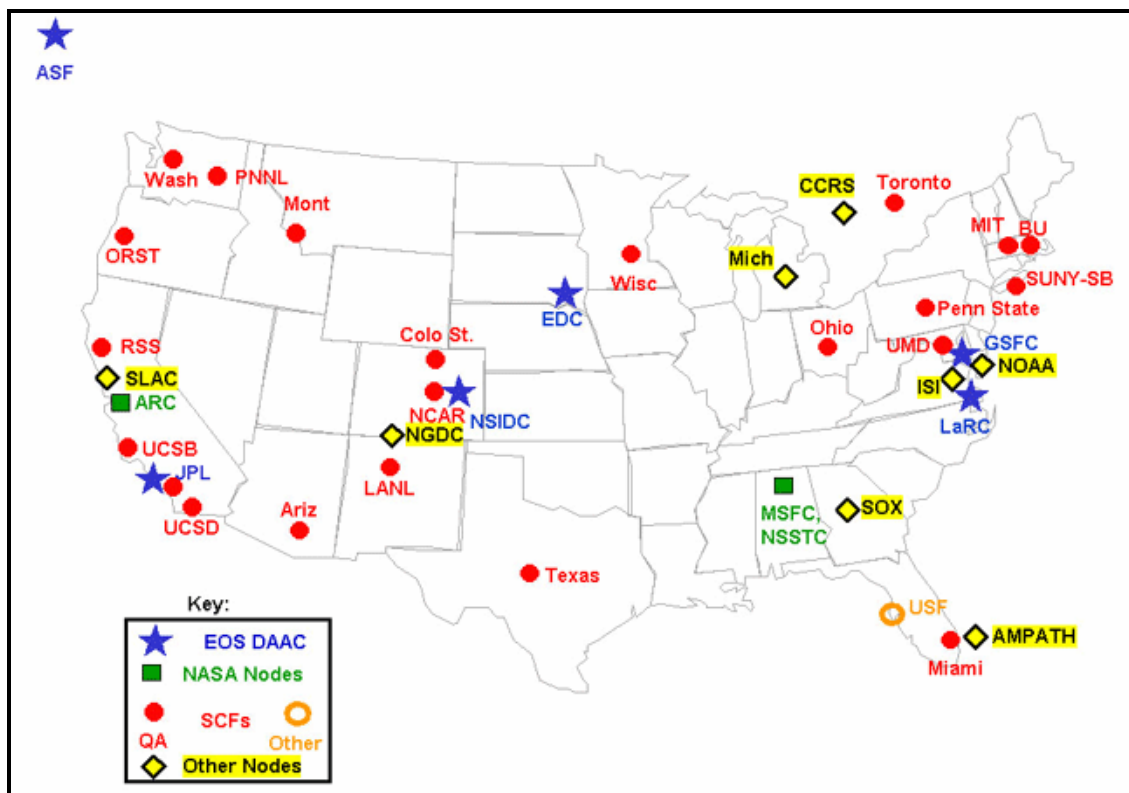


Figure 1. Domestic Sites

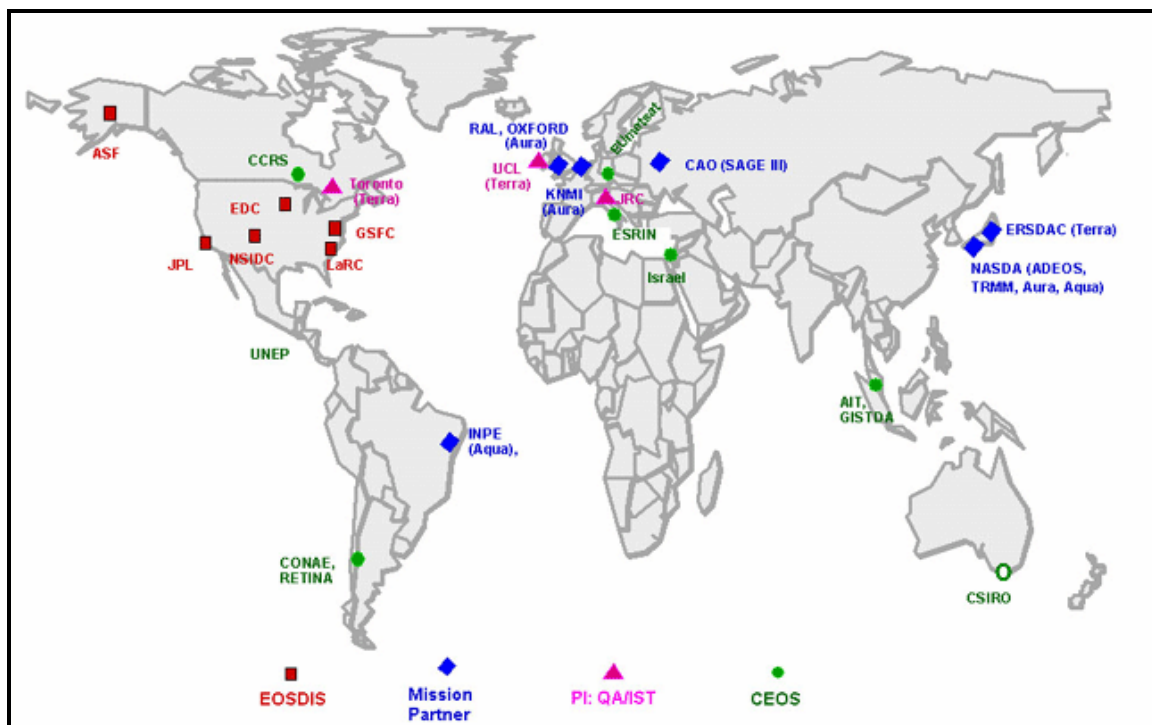


Figure 2. International Sites

Passive metrics include the amount of data flowing over various time periods, for specific circuits, interfaces, protocols, sources and destinations. Other metrics can relate to the number of various types of error conditions. Active metrics focus on throughput capability, and also include round-trip time (RTT), packet-loss percentage, and the number of hops in the route. They are measured end-to-end, from one end system to another, with little knowledge of the network in between (unless intermediate nodes along the path also participate in the active testing).

At the top level of the ENSIGHT performance-measurement system, there are the active-data-collection and passive-data-collection subsystems, which make and collect the measurements. The results are sent to a database subsystem for storage. A set of graphing programs extracts data from the database, and produces graphical displays. In addition to graphs showing either just active or just passive results, it has been found useful to combine related active and passive measurements into "integrated" graphs. The graphs are integrated into web pages, and sent to the web server; the web pages are accessed through a web proxy server.

The ENSIGHT system is hosted on an i686 dual-processor-based computer with 60 Gigabyte disk space running Red Hat Linux 7.0. With the exception of the Oracle database engine, the ENSIGHT system has been developed largely using open-source software components. All database accesses however are standard SQL and an open-source database (e.g., MySQL) could easily be used instead. The system is generally driven by Perl scripts, which are often executed through the use of *cron* jobs. Some of the open-source products include Perl v5.6.1, SourceForge's Net::SNMP package [6], Lincoln D. Stein's GD::Graph package [7], the DBD and DBI database interface packages [8] by Tim Bunce, Mark Fullmer's Flow Tools [9], and the Apache web server [10].

Graphs of active measurements, passive measurements, and integrated measurements can all be accessed from the ENSIGHT web site at <http://ensight.eos.nasa.gov>. Passive measurements that involve proprietary IP addresses are provided on a separate web server and are accessible only to authorized EOSDIS networking personnel. A comprehensive view of network performance can be obtained from the combination of these techniques. The different types of results available from ENSIGHT are described in more detail in the remainder of this paper.

Section 2 describes the passive-measurement subsystem. Section 3 addresses active measurement, while Section 4 addresses integrated measurement. In all three of these sections we present specific techniques used for performance measurement and discuss the types of information that can be gleaned from these measurements. Section 5 concludes the paper.

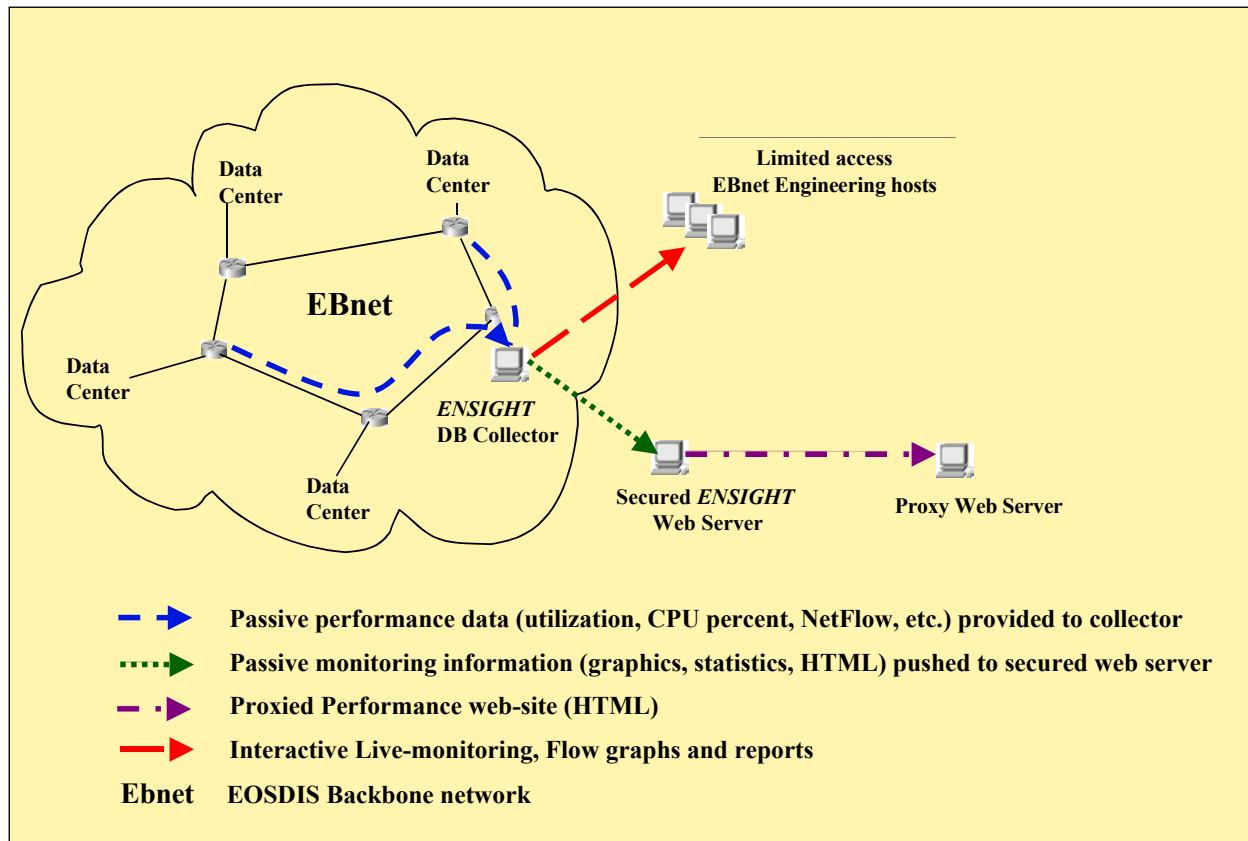


Figure 3. Passive-Measurement System Overview

2. Passive Measurements

Passive measurements are derived from information stored on network devices and are either polled from the database collector periodically, or pushed to the collector from the devices themselves. An overview of the passive-measurement subsystem is shown in Figure 3. The passive measurements fall into two categories: Simple Network Management Protocol (SNMP) objects data and NetFlow [11] flow-statistics data. Each of these categories is discussed below.

2.1 Object Measurements

This data primarily consists of SNMP interface input and output byte counters. The gathered data is manipulated into graphs that depict interface load over time. This is of course a very standard report, e.g., Multi Router Traffic Grapher (MRTG) [12], and we have modeled the collection and round-robin storage after two very capable open-source toolsets; the MRTG and the Round-Robin Database [13] developed by Tobi Oetiker. This capability was redeveloped in order to store this data into the Oracle database and have it readily accessible for future tool developments or enhancements.

Device: ecs_edc
Object: 1.3.6.1.2.1.31.1.1.1.6.12

OID Name (30S):	ge600_iHCInOctets	OID Type (12S):	COUNTER
Label (10S):	ge600in	Description (30S):	In from vBNS
IP Address (15S):	192.168.50.2	Archive Method (8S):	NORMAL
Collection Interval (5N):	5	Collection Offset (2N):	0
Graphing Interval (5N):	30	Graphing Offset (2N):	0
Graphing Partner (40S):	1.3.6.1.2.1.31.1.1.1.10.12	Graphing Multiplier (16F):	8
Use Maximum? (1S):	Y	Maximum Value (12N):	1250000000
Legend (30S):	Traffic In	Vertical (20S):	Bits/Second
Last Observ Time (N/A):	09/24/2003 12:55:03	Last Observation (N/A):	340052734109799

Figure 4. Managing an Object

Data can be collected for any available SNMP object. Currently measurements such as interface utilization, router CPU usage, and router memory availability are tracked. The toolset provides a web-based user interface for adding or modifying tracked objects. A sample is shown in Figure 4.

This figure lists attributes of a particular SNMP object. Several attributes are of interest. The collection interval indicates how often the device should be queried for this object; the graphing interval indicates how often a new graph should be created from the collected data; the graphing partner indicates whether a second object should be graphed on the same graph and what it is (this is useful for comparing inbound and outbound interface utilization.)

The collection offset permits objects to be collected at different times in order to spread out the load on the collector host resources; same with the graphing offset. Otherwise objects are collected every collection interval from 12:00 midnight. The maximum value is useful for preventing spikes that would dominate the graph. The OID type indicates whether the object is tracked as a counter or a gauge. Counter objects are reduced to gauge readings upon input.

The archive method of "NORMAL" will produce 4 MRTG-like graphs showing data collected every 5 minutes, and then averaged over longer periods of time (half-hour, 2 hours, 24 hours). Figure 5 below shows a sample of the 5-minute graph, while Figure 6 shows averages calculated over the maximum 24-hour time period.

Device: ecs_edc
Object: ge600 vBNS

'Daily' Graph (5 Minute Actuals)

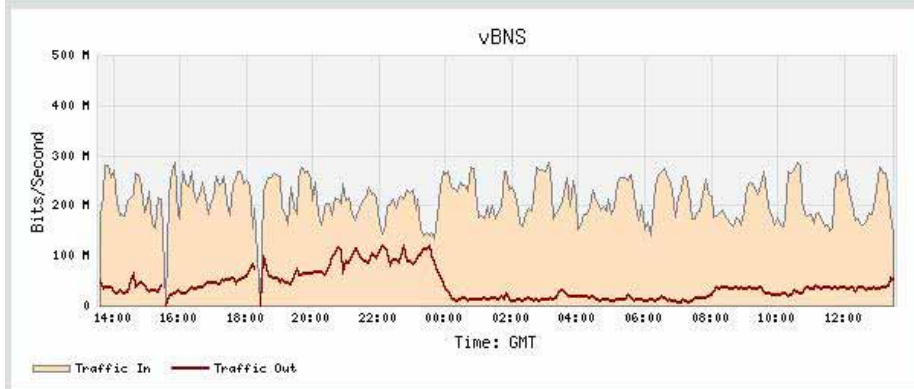


Figure 5. 5 Minute Averages

'Yearly' Graph (1 Day Averages)

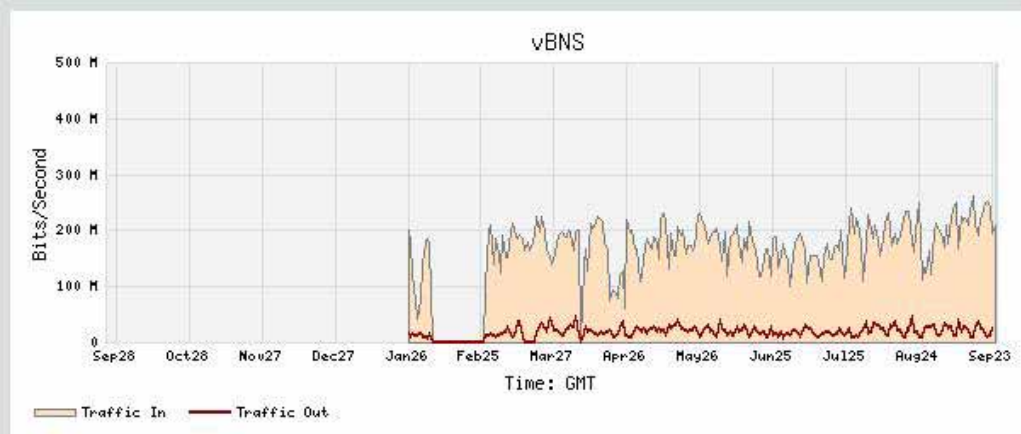


Figure 6. 24 Hour Averages

Note that Figure 6 indicates a gradual increase in daily utilization and that it is butting up against the 300 Mbps ATM VBR PVC (Asynchronous Transfer Mode Variable Bit Rate Permanent Virtual Channel) limit for this circuit (also seen in Figure 5). A graph like this becomes useful for capacity planning, indicating the potential need to increase the ATM circuit contract.

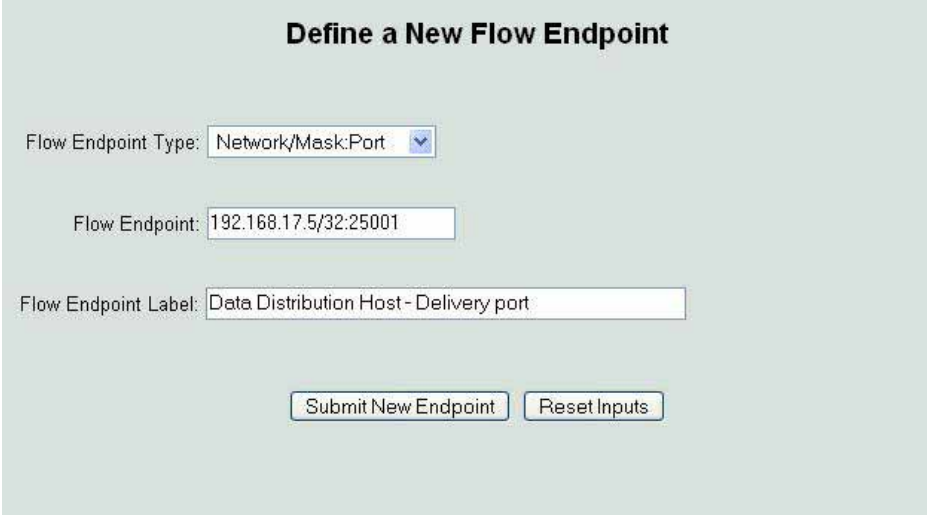
2.2 Flow Measurements

The flow-measurements portion of the passive-measurements capability collects and graphs data pertaining to specified flows through the EOSDIS system. This information is useful for tracking particular contributors to overall network usage. The capability is provided to track flows from as specific as host/port to host/port to as large as an aggregate of data transported from one DAAC to another. In addition, daily reports are generated which identify the individual components and their contribution to the aggregate flows. Finally a web-based interface permits the user to interrogate the stored NetFlow data in a variety of ways.

This capability is heavily dependent on the excellent Flow Tools suite developed by Mark Fullmer. Flow Tools captures NetFlow data exported from devices throughout the network and stores this data for several months at a time. The length of time NetFlow data is archived is merely dependent on available storage and the quantity of data exported from the devices. The Flow Tools suite offers a myriad of ways of interrogating the stored NetFlow data.

A flow is identified by its two endpoints. In order to begin tracking a particular flow the endpoints must first be defined and assigned to the flow. Figure 7 below shows the web page provided for defining an endpoint.

In Figure 7 a Network/Mask:Port type of endpoint is defined. Other types of endpoints that can be defined include Network/Mask, Interface, Interface:Port, and Autonomous System. Once both endpoints of a flow have been defined (some endpoints can be re-used) a flow can now be defined (Figure 8.)



The screenshot shows a web form titled "Define a New Flow Endpoint". It contains three input fields: "Flow Endpoint Type" with a dropdown menu set to "Network/Mask:Port", "Flow Endpoint" with the text "192.168.17.5/32:25001", and "Flow Endpoint Label" with the text "Data Distribution Host - Delivery port". At the bottom of the form are two buttons: "Submit New Endpoint" and "Reset Inputs".

Figure 7. Entering a New Flow Endpoint

The defined flow in Figure 9 is intended to collect flow data from the Goddard Space Flight Center DAAC in Maryland to the EROS Data Center (EDC) DAAC in Sioux Falls, SD. Note that the capability to exclude certain information (in this case port 5500 which actually is the port for the active performance testing (PT) measurements described in Section 3) is provided by using a minus sign. The Flow Status provides several options including Inactive, Collect Only, Collect and Graph, or Daily Report.

Create a Flow for Collection on: ecs_nsidc

Source Endpoint:

Destination Endpoint:

Flow Group:

Flow Status:

Figure 8. Creating a New Flow

Modify a Flow
Device: ecs_gsfc

Source Type: NW
Source Endpoint: 192.168.220.0/24
Source Label: DAAC - GSFC Production LAN

Destination Type: NP
Destination Endpoint: 172.16.4.0/22:-5500
Destination Label: DAAC - EDC Production LAN (no PT)

Flow Group:

Flow Status:

Figure 9. A Defined Flow

Once the data is collected for a flow both quantity and rate graphs are produced, four of each, similar to the Object tracking graphs described above. Flow data may pass through several network devices, and the user can use discretion to pick the appropriate device from which to collect the data. An 'All Devices' option is under development, which would permit, for example, collecting 'all data initiating from multiple devices throughout the system but destined to a particular external host.'

Figure 10 provides an example of a flow-quantity graph. Here the graph shows daily quantities for the flow defined above. In this case, the GSFC DAAC is flowing over 2 Terabytes of data per day to the EDC DAAC. Figure 11 shows the ENSIGHT system's ability to look at utilization rates of specific flows. The graph looks very much like a standard interface-utilization graph but is limited to a specific flow. Figure 12, another rate graph, tracks the delivery of data from the EDC DAAC to the University of Maryland.

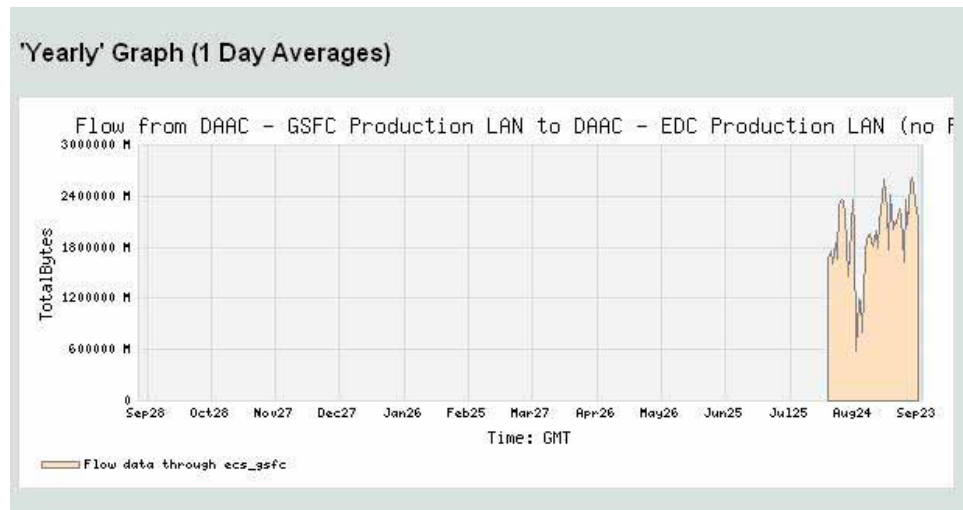


Figure 10. Flow Quantity Yearly

Flow Rates

'Daily' Graph (5 Minute Actuals)

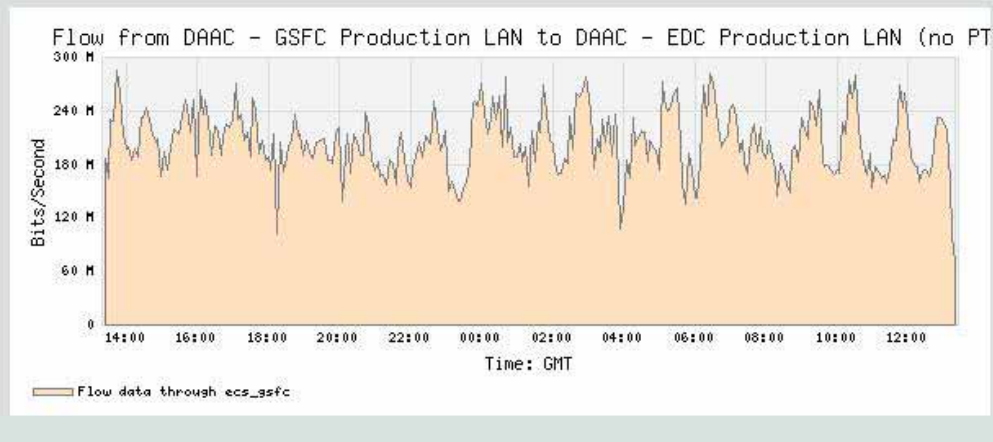


Figure 11. Flow Rate, 5 Minute Interval

'Weekly' Graph (30 Minute Averages)

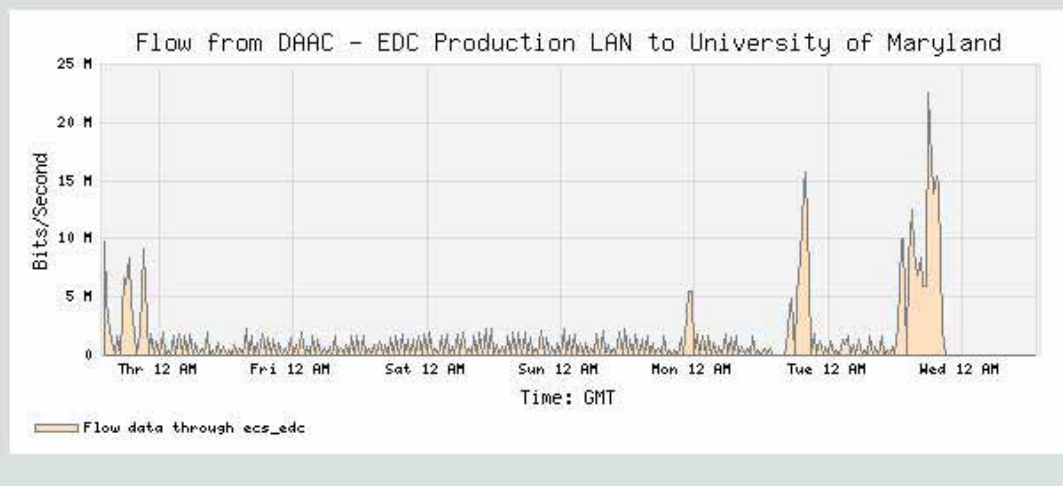


Figure 12. Flow Rate to University of Maryland

Figure 13 shows a typical daily report. The daily report is generated once a day and breaks out the specific contributors to the flow that is being collected and for which daily quantities and rates are graphed. In the figure, the [hyperlink \(underlined\)](#) provides a quick link to the flow quantity and rate graphs. The option to look at another day is



Figure 13. Daily Report

provided as well. The sustained rate over the 24-hour period for this flow was 268 Mbps. Note the 2-Terabyte daily transfer is between two specific hosts.

Figure 14 shows the web page used to build custom flow reports. This page is primarily a front-end to the Flow Tools capability customized for the EOSDIS network. It exploits the great reporting flexibility of the Flow Tools suite. This tool permits a network analyst to closely examine any aspect of the flow of data through the network over any time period for which the NetFlow data remains stored. Options exist to examine a view as broad as all flows between two interfaces to (for example) a very narrow examination of individual flows (with time values) between two host:ports over a specified interface for 3 seconds. An option is provided to report all hosts with DNS resolved names, as well as the option to sort the resulting report on a particular column. Specific data can be excluded by preceding any form input with a minus sign (-). The report is created quickly owing to the efficiency of the underlying Flow Tools.

A sample report is shown in Figure 15. This report captured all of the active-test measurements passing through the ecs_larc router between 10:00:00 and 11:00:00 on September 24, 2003.

Passive Monitoring - Custom Reports

Filter Criteria:

Device:

Start Date: (e.g., 7/17/2003) Start Time: (e.g., 11:26:00)

End Date: (e.g., 7/17/2003) End Time: (e.g., 11:26:00)

Source IP: (e.g., 192.168.16.0/22) Source Port: Source Interface: Source AS:

Dest IP: (e.g., 0.0.0.0/0) Dest Port: Dest Interface: Dest AS:

Report Type:

Statistics: Printed:

Sort Field: Cutoff Lines: Resolve Addresses:

Figure 14. Building a Custom Flow Report

Powered by Mark Fullmer's Flow Tools Suite!

Report Parameters:

Report: Source/Destination IP Sort Field: 4 Start: September 24, 2003 10:00:00

Device: ecs_larc Lines Cutoff: 100 End: September 24, 2003 11:00:00

Source Port: Destination Port: 5500

#	# Source	Destination	flows	octets	packets
#	somelarchost.nasa.gov	xxxx.essc.psu.edu	2	108299680	76276
	somelarchost.nasa.gov	xxxx.bu.edu	2	105783524	71101
	somelarchost.nasa.gov	xxxx.cals.arizona.edu	2	85956508	57775
	somelarchost.nasa.gov	xxxx.scd.ucar.edu	2	70573628	47439
	somelarchost.nasa.gov	xxxx.rsmas.miami.edu	2	61370260	41100
	somelarchost.nasa.gov	xxxx.lanl.gov	2	21124940	14205
	somelarchost.nasa.gov	xxxx.physics.utoronto.ca	1	6341264	4266
	somelarchost.nasa.gov	xxxx.ge.ucl.ac.uk	3	4487962	3065
	anotherlarchost.nasa.gov	xxxx.jpl.nasa.gov	6	720	12
	anotherlarchost.nasa.gov	xxxx.jpl.nasa.gov	4	480	8
	anotherlarchost.nasa.gov	somestsfhost.nasa.gov	4	480	8
	anotherlarchost.nasa.gov	somensidhost.nasa.gov	2	240	4

Figure 15. Custom Flow Report

The Custom Flowgraph tool provides the network engineer with the ability to visually analyze network traffic behavior. Figure 16 shows the results of a query that examined a one-hour period, looking at FTP transfers (TCP port 20.) This particular tool was used to demonstrate that the EOSDIS network was not the cause of recent FTP file transfer problems when a flowgraph was generated showing both FTP transfers and active-performance-measurement traffic during the same time period. It was clear that the active TCP measurements were able to use all of the available bandwidth, but the FTP transfers were otherwise constrained.

Of course exporting NetFlow data continuously from multiple devices is not free. The capability to track the network impact caused by the NetFlow data export is also provided with ENSIGHT. Periodic samples of traffic on the LAN that is attached to the collector device are collected and graphed. The impact on EOSDIS networks created by collecting NetFlow data from the current four network devices (to increase soon to nine devices) is minimal as is seen in Figure 17.

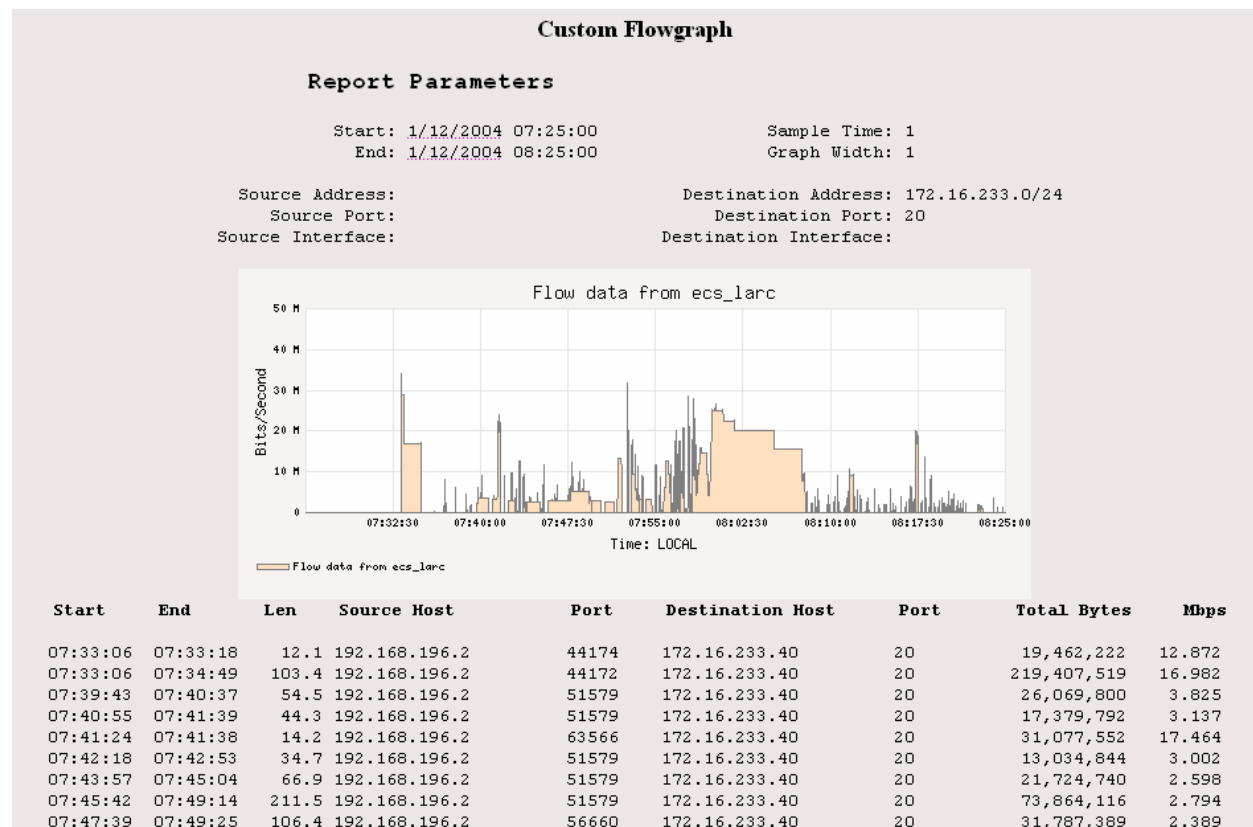


Figure 16. Custom Flowgraph

NetFlow Load on Collector Host LAN

These graphs were last updated: 01/13/2004 12:08:14 (Local)

'Daily' Graph (5 Minute Actuals)

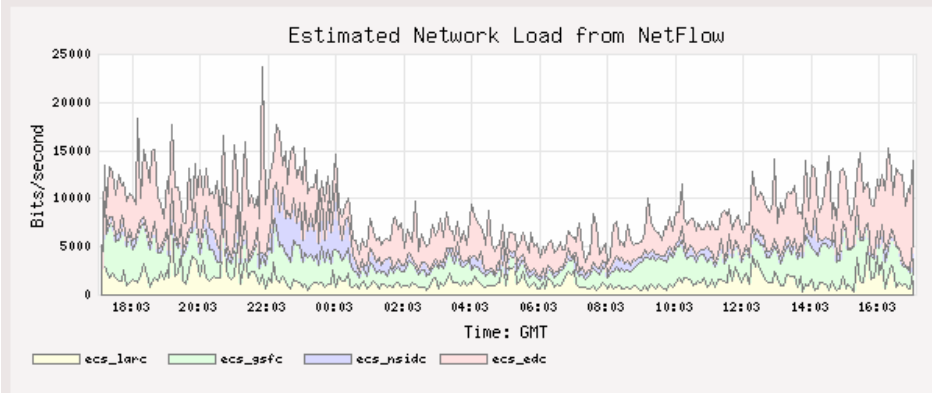


Figure 17. Custom Flow Report

2.3 Live Monitoring

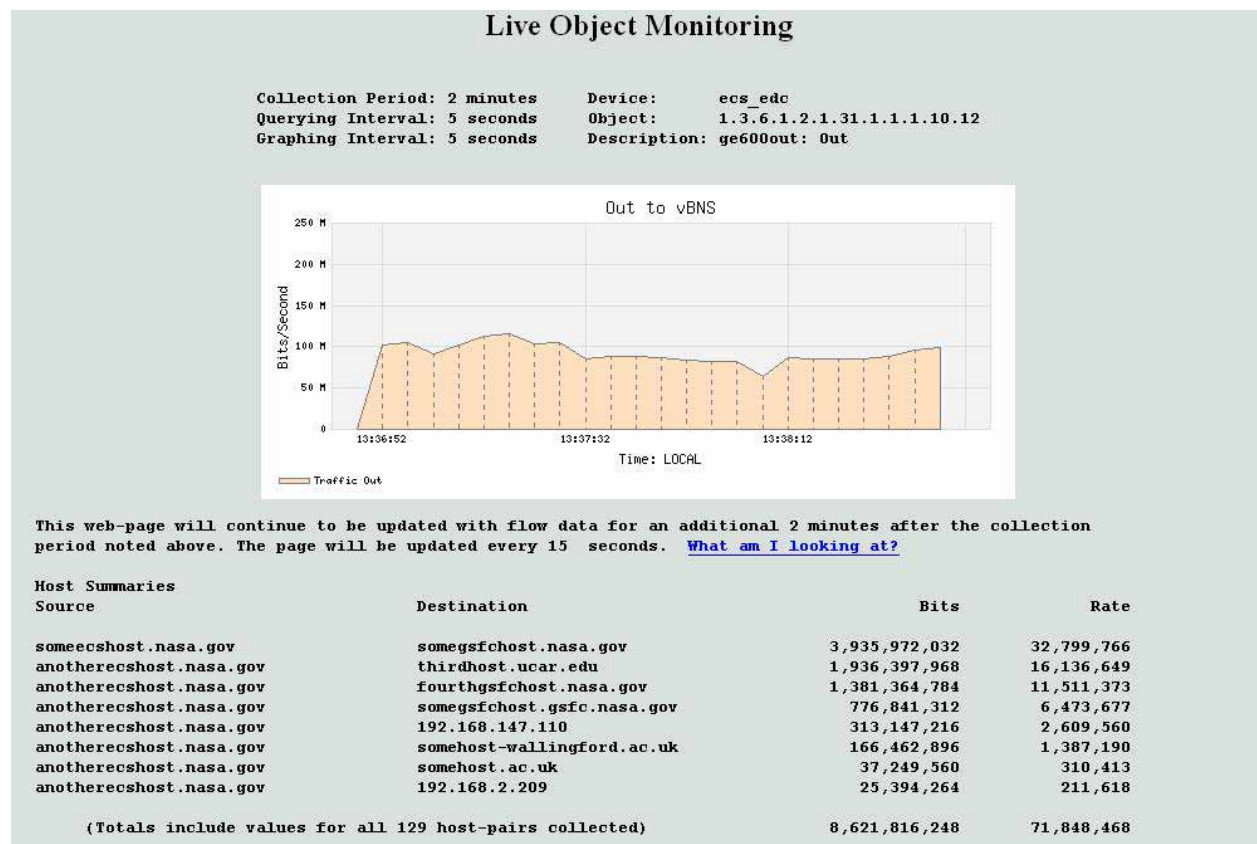
The SNMP object-data-tracking capability and the flow-measurement capabilities have been merged to provide ENSIGHT's live-monitoring capability. This function provides the user with a 'live' look at an interface together with the flows that are passing through that interface during the monitoring period. The user selects an interface for monitoring via a web page. The ENSIGHT software begins pooling the SNMP interface-byte-count object every 5 seconds (this is the minimum rate at which network devices tend to update their SNMP counters.) A report (see Figure 18) is produced and updated automatically every 5 seconds while the user is viewing it.

There are three components to this report. The first is the typical interface-utilization graph but which is now presenting data at a much finer granularity than the 5 minutes used in the SNMP object-tracking graphs discussed above. The second section is created from live NetFlow data and shows host-to-host activity. The third component is like the second but provides additional port information. The information in the second and third sections is derived from the live collection of NetFlow data and is posted to the updating web page as it becomes available.

Because NetFlow data is not exported from the routers until the flow expires, the information in the flow sections trails in time that of the utilization graph. Thus, in Figure 18, we see that the utilization graph shows an average of approximately 85 Megabits per second, while the total of flow data acquired so far is only 71 Megabits. To correct this, the page is left to update for several minutes after the live utilization tracking is

ended. As a result, in terms of rates, after a few minutes both sections are in much closer harmony.

The third section, which includes port information, has a Flow Rate and an Overall Rate column. The flow rate is that rate achieved during the time that the flow existed. The overall rate represents the total data through the flow, which may be a shorter period, divided by the total period for which utilization data was collected. An examination of the data in the third section illuminates how EOS computers use multiple, parallel TCP streams to accomplish large-scale data transfers quickly.



Individual Flows Source	Port	Destination	Port	Bits	Flow Rate	Overall Rate
someecshost.nasa.gov	28515	somegsfchost.nasa.gov	20	1,723,516,712	33,744,159	14,362,639
someecshost.nasa.gov	28204	somegsfchost.nasa.gov	20	1,721,676,848	35,242,709	14,347,307
anotherecshost.nasa.gov	54635	192.168.147.110	4158	312,395,232	3,031,080	2,603,293
someecshost.nasa.gov	27297	somegsfchost.nasa.gov	20	256,529,160	26,484,530	2,137,743
someecshost.nasa.gov	28988	somegsfchost.nasa.gov	20	232,838,568	30,144,817	1,940,321
anotherecshost.nasa.gov	56097	thirdhost.ucar.edu	60792	213,753,880	8,345,848	1,781,282
anotherecshost.nasa.gov	56074	thirdhost.ucar.edu	60790	212,765,704	7,916,568	1,773,047
anotherecshost.nasa.gov	55896	thirdhost.ucar.edu	60784	212,752,888	7,880,904	1,772,940
anotherecshost.nasa.gov	55989	thirdhost.ucar.edu	60786	212,748,024	8,105,304	1,772,900
anotherecshost.nasa.gov	55845	thirdhost.ucar.edu	60782	212,747,352	8,564,708	1,772,894
anotherecshost.nasa.gov	55720	thirdhost.ucar.edu	60780	212,728,856	8,803,544	1,772,740
anotherecshost.nasa.gov	56015	thirdhost.ucar.edu	60788	212,716,104	8,255,049	1,772,634
anotherecshost.nasa.gov	55671	thirdhost.ucar.edu	60778	204,463,720	8,797,165	1,703,864
anotherecshost.nasa.gov	56111	fourthgsfchost.nasa.gov	37330	196,390,376	11,289,398	1,636,586
anotherecshost.nasa.gov	55734	fourthgsfchost.nasa.gov	37325	196,377,960	11,664,169	1,636,483
anotherecshost.nasa.gov	56020	fourthgsfchost.nasa.gov	37328	195,668,272	11,767,396	1,630,568
anotherecshost.nasa.gov	56068	fourthgsfchost.nasa.gov	37329	194,954,024	11,747,048	1,624,616
anotherecshost.nasa.gov	55848	fourthgsfchost.nasa.gov	37326	194,942,816	11,551,482	1,624,523
anotherecshost.nasa.gov	55969	fourthgsfchost.nasa.gov	37327	193,579,072	11,709,355	1,613,158
anotherecshost.nasa.gov	55725	somehost-wallingford.ac.uk	33524	160,605,896	1,398,482	1,338,382
anotherecshost.nasa.gov	56155	thirdhost.ucar.edu	60794	133,384,464	8,183,095	1,111,537
anotherecshost.nasa.gov	56189	fourthhost.nasa.gov	37331	121,682,320	11,492,474	1,014,019
anotherecshost.nasa.gov	55649	fourthhost.nasa.gov	37324	87,742,960	11,529,955	731,191
anotherecshost.nasa.gov	56193	mopl.eos.ucar.edu	60796	71,567,448	7,146,025	596,395
anotherecshost.nasa.gov	55977	fifthhost.nasa.gov	38807	48,533,384	7,430,095	404,444
anotherecshost.nasa.gov	56064	fifthhost.nasa.gov	38820	48,460,080	7,035,435	403,834
anotherecshost.nasa.gov	55949	fifthhost.nasa.gov	38805	48,450,616	11,330,826	403,755
anotherecshost.nasa.gov	56024	fifthhost.nasa.gov	38814	48,450,592	11,373,378	403,754
anotherecshost.nasa.gov	56010	fifthhost.nasa.gov	38812	48,450,592	10,785,973	403,754
anotherecshost.nasa.gov	55999	fifthhost.nasa.gov	38810	48,450,272	11,535,779	403,752
anotherecshost.nasa.gov	56052	fifthhost.nasa.gov	38818	48,436,128	11,243,298	403,634
anotherecshost.nasa.gov	56187	fifthhost.nasa.gov	38834	48,436,104	11,348,665	403,634
anotherecshost.nasa.gov	56140	fifthhost.nasa.gov	38830	48,436,056	11,434,385	403,633
anotherecshost.nasa.gov	56031	fifthhost.nasa.gov	38816	48,435,776	10,792,285	403,631
anotherecshost.nasa.gov	56161	fifthhost.nasa.gov	38832	48,435,704	11,348,571	403,630
anotherecshost.nasa.gov	56103	fifthhost.nasa.gov	38824	48,425,432	11,573,955	403,545
anotherecshost.nasa.gov	56091	fifthhost.nasa.gov	38822	48,425,424	11,518,892	403,545
anotherecshost.nasa.gov	56109	fifthhost.nasa.gov	38826	48,425,424	11,663,156	403,545

Figure 18. Live Interface Monitoring

3. Active Measurements

The active-measurement system currently utilizes about 30 source nodes, and 100 sink nodes. The sink nodes generally are passive; they run servers, which respond to requests from clients at the source nodes. The source nodes clients are invoked by *cron* jobs. Figures 1 and 2, presented earlier in the paper, show the participating sites. Figure 19 is an overview of the active-measurement system.

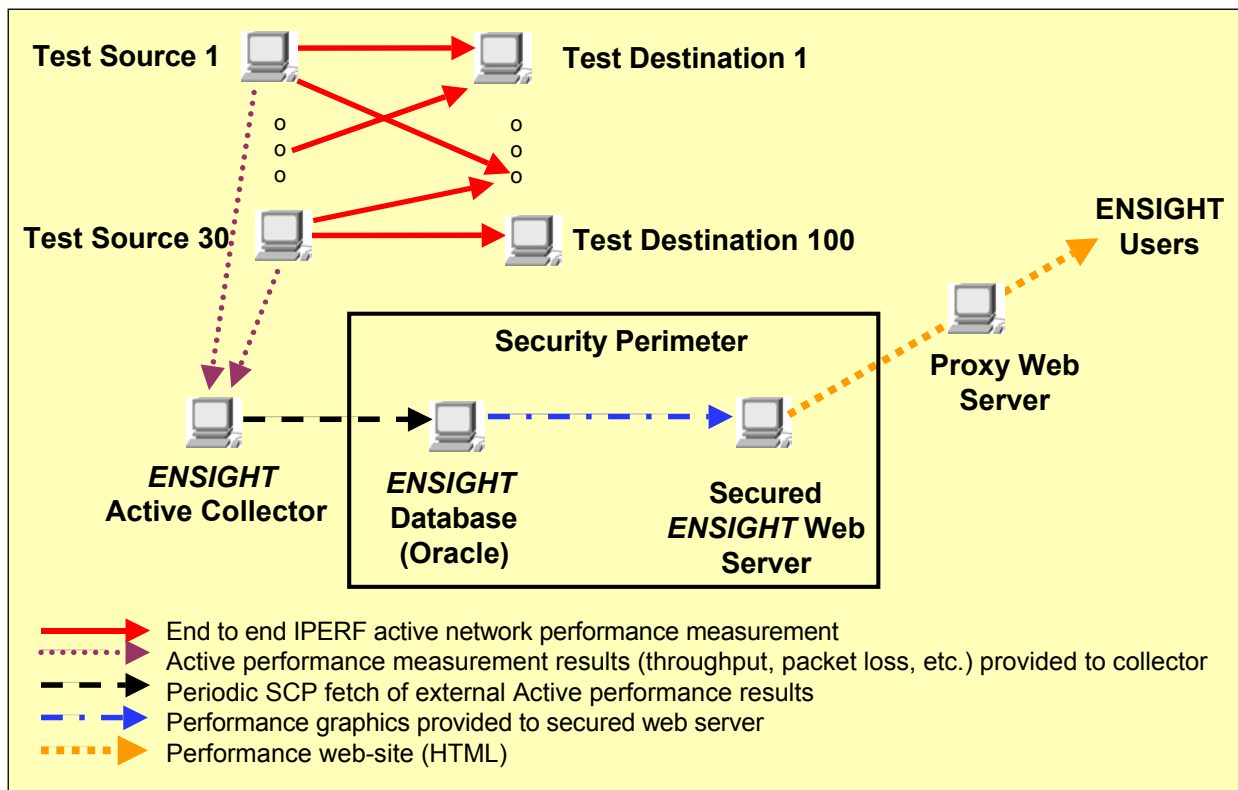


Figure 19. Active-Measurement System Overview

3.1 Overview of Source Nodes and Active-Measurement Tests

The source nodes actively drive the tests; testing to each destination is initiated hourly by a *cron* job. The test times are scheduled throughout each hour in order to avoid overlapping with other tests from the same source, and also with other tests to the same destination from other sources.

In the nominal case (there are many variations), the first test step is a traceroute to the destination. From this traceroute a number of hops will be derived – the last hop to respond will be counted if the destination is not reached. Also, the RTT will be derived, provided the intended destination is reached. This will be used as the RTT if the ping test does not provide one.

Next is the standalone ping test. In this test, 100 pings are sent to the destination, and the RTT and the number not returned is extracted. If obtained, this RTT will be used in preference to the one found in the traceroute. The loss from this ping test (if under 100%) will be extracted as packet loss if there is no concurrent ping test.

Next is the throughput test, and on some nodes another "concurrent" ping test. The throughput test is designed to last for 30 seconds, so the concurrent ping test is used on nodes which are enabled to send 100 pings in this period. If so, the packet loss is extracted from the concurrent ping test.

The throughput test is one of four types, depending on characteristics of both the source and destination nodes. *Iperf* [14] (available from the National Lab for Advanced Network Research (NLNR [15])) and *nuttcp* [16] (NASA developed) are preferred, due to their rich set of features. An older program *tcpwatch* [17] (also NASA developed) has been mostly phased out, but is still in use on some systems. Note that *tcpwatch* and *iperf* are compatible (i.e., the client for one works with the server for the other, except for their default port numbers) and get similar results. Whichever of these three programs (all derived from *ttcp*) is used, the tests are set up to keep the TCP send buffer full for 30 seconds.

Alternatively, if none of the above programs can be used, *ftp* is the fallback throughput test. In this case a file size is chosen to target the transfer time to about 30 seconds, although this can change as the network is upgraded. Accordingly, no concurrent ping test is used with *ftp* throughput tests – because it can't be assured that the pings are actually concurrent!

Note that these tests all use the source machine's standard TCP stack, and thus because of TCP's congestion-control mechanism will tend not to allow the network to be swamped by test packets. This would not be true of UDP tests, which could easily be configured to exceed the network's capacity. But TCP will tend to share the network more or less equally among all TCP streams, so other users will still get a share of the available throughput. This, and the short (30 seconds per hour) test duration, has enabled these tests to continue in an ongoing way along with user operational flows. The impact on users is negligible.

All the data extracted from these tests is collected on the source machine in a "results" file, which is sent on an hourly basis to a "collector node," where it is ingested into the database. These results are then plotted and displayed on the web site, as described in Section 3.4. In case of the inability to send the data, it is retained on the source node, and new results appended to the results file. An attempt to send the data is made hourly; the accumulated data is deleted only after it is successfully sent.

3.2 Sink nodes

Ideally, a sink node will include the capability to respond to pings, traceroutes, and one of the four throughput programs listed above. In most cases the throughput server is configured to be available continuously, but an effort is in progress to make the server available on a schedule corresponding to the source-node schedule.

3.3 Navigating the Active-Measurement Web Pages

The active-measurement web pages and displays are organized on a "destination" node basis. There are several groupings of destinations that can be accessed and used for individual site selection.

Maps. The U.S. or International maps can be selected, which then shows a map of all of the sites tested in those categories. Clicking on an individual site on these maps will lead to the "destination page" for that site. If multiple systems are tested at the same location, clicking on that site will link to a page with a list of systems tested at that site. That page will then link to the individual destination pages.

EOS Mission sites. Selection of one of the EOS Missions will link to a map showing the sites tested which participate in that mission. Figure 20 is a sample mission map for the "Terra" mission. On the map is a tiny graph for each site, located in the approximate geographical map location for that site. These graphs show the daily min-max range for the throughput testing to that site for the last week, the median for each day, and the requirement against which the throughput is compared, if any. If there is more than one requirement, only the highest one is shown. If there is more than one source testing to a site, only one source is shown. This will normally be the source corresponding to the requirement. The border color of these tiny graphs, which are updated hourly, is used to indicate the status of the performance relative to the requirement, according to the chart that appears on the page. In this way, the status of all nodes associated with a mission can be evaluated in a short glance.

From these mission pages, clicking on any of the tiny graphs will link to the "destination page" for that site. This can also be achieved by clicking on the name of the site from the list at the left or bottom of the page. Links to other categories of sites can be found at the top of the page.

Network sites. Two categories of networks can be selected from the Active-Measurements page: EMSnet, and Research Networks. EMSnet is the EOS internal production network. This link is available only to authorized users, and is intended as a network-monitoring tool. It uses tiny graphs to show the status of all the nodes tested on the EMS network. The "Research Networks" (RENs) page lists sites at hubs of various Research Networks that are participating in these tests.

Organization sites. These are sites grouped organizationally – sites belonging to participants in the Committee on Earth Observing Satellites (CEOS) [18], the DAACs, sites participating in the Earth Science Technology Office (ESTO) [19] Computational Technologies Project [20], and a large number of nodes located at NASA GSFC.

Other. Finally, "Other" is a list of a few sites that don't fit any of the above categories.

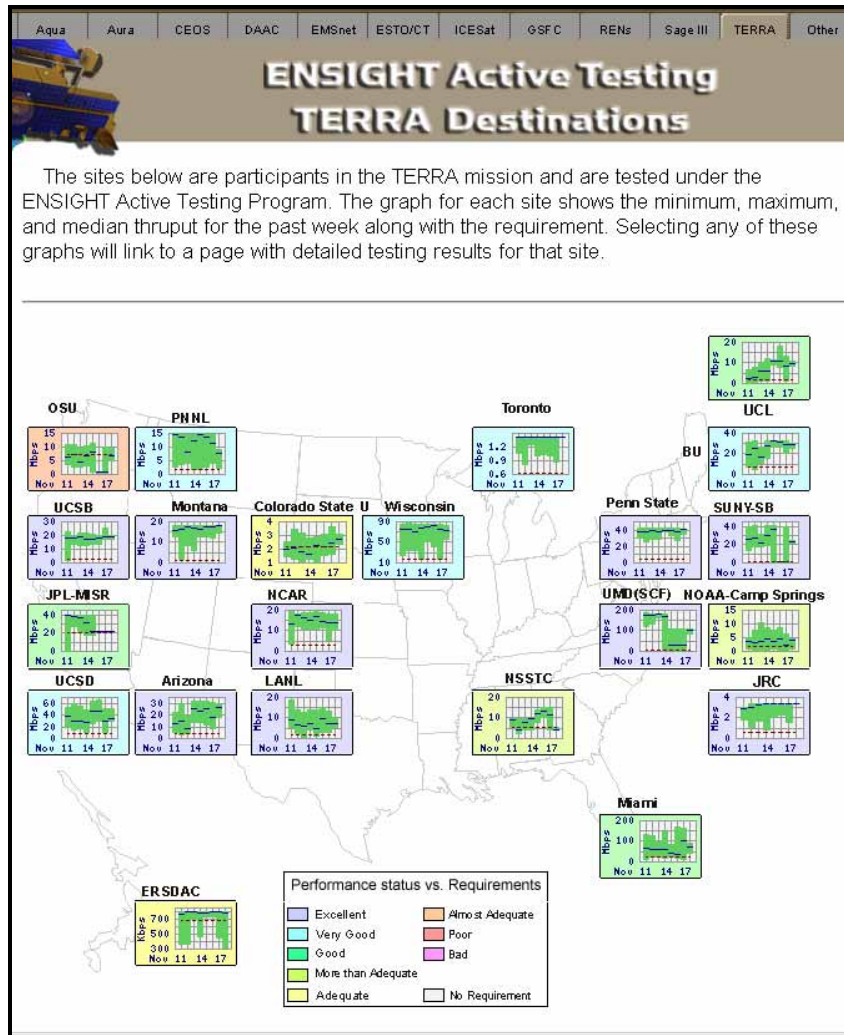


Figure 20. Terra Mission – Clickable Map

3.4 Destination-Page Display

Selecting a destination, either from a map page or a list, links to a destination page. Figure 21 is a sample destination map for Pennsylvania State University.

The destination page mainly consists of up to 16 small graphs, showing the results of measurements to that site. Note that all tests shown on graphs on destination pages are TO that site. The graphs are arranged in a 4 x 4 array, and can all fit on most screens at the same time. The leftmost column contains "Throughput" graphs. To the right of those are "Loss" graphs, next are "Hops" graphs, and on the right are "RTT" graphs.

The top row of these graphs shows individual results for each of the four parameters for the current week: today plus the previous 7 full days. The second row is based on the

same data, but shows the median for each hour of the day (in Greenwich Mean Time (GMT)). In this way time-of-day sensitivities can be easily seen, and the variability between individual tests is somewhat smoothed out. The requirements (if any) are shown on the throughput graphs as dashed lines. The third row shows a longer period (4 full months plus the current month), displaying daily median values. The fourth row shows weekly medians for the current year plus 2 full years in the past.

The line colors on these graphs indicate the various source nodes that test to that destination. Source colors are consistent throughout all destinations (source colors have been chosen with the intent of generating readable graphs, although with the various permutations of sources and destinations, some graphs have some similar colors).

Clicking on any of the smaller graphs links to a larger version of the same graph. The source key is not shown on the small graphs, but can be found on the large graphs. Some additional information is also shown on the destination pages. Typically included are a description of the node, and the current status of the throughput versus the requirements. Below the graphs is an indication of the route used from the various sources to the destination.

Throughput graphs. The throughput graphs are based on the throughput program run, either *iperf*, *nuttcp*, *ftp*, or *tcpwatch*. *Iperf* and *nuttcp* are capable of running multiple parallel TCP streams. This option is often used when the throughput of a single stream is limited not by the network, but instead by the maximum TCP window size of one of the end systems (or sometimes by a proxy firewall). If so, the throughput shown on the graphs is the sum of all the parallel streams. This condition is noted in the footer section of the destination page.

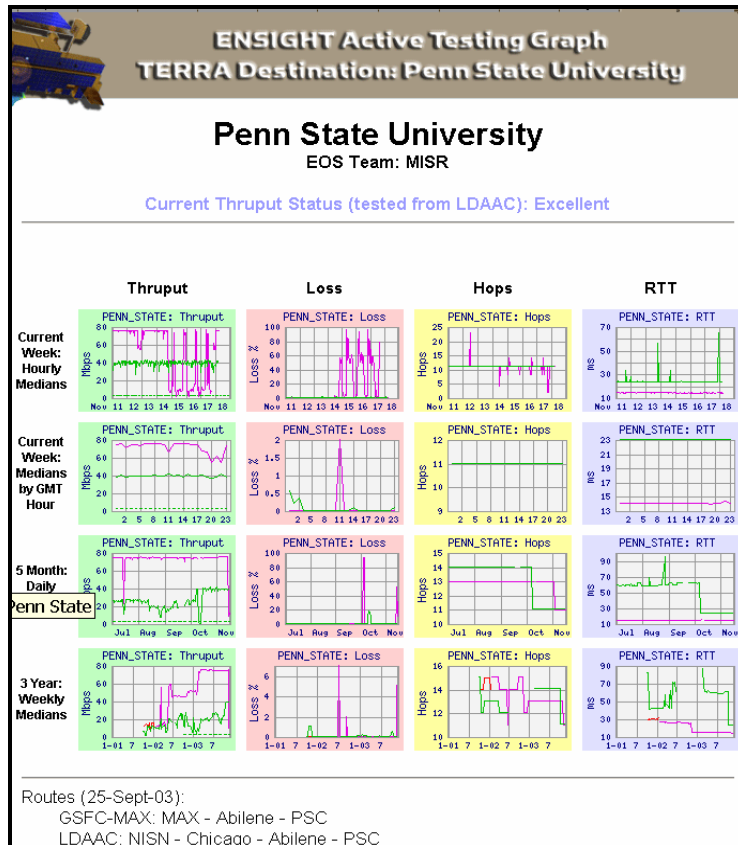


Figure 21. Destination Page: Penn State University

Hops graphs. The hops graph is based on the results of a traceroute. The intent is to be able to see when route changes occur. (Note: some route changes have the same number of hops. This will not be seen as readily, but perhaps the RTT graph will show this occurrence.) The graph shows the hop count for the last node which responds to the traceroute – the “* *” responses are ignored. The important note here is that this may or may not be the actual destination. This presentation was favored over always showing the maximum number of hops (30) when the destination did not respond, based on the fact that many campuses block ICMP inside their campus. In this case the hops count will indicate the number of hops to the campus edge, and will be useful in determining route changes from the source node to the campus edge. So there is no clear indication on this graph whether the end node is or is not reached.

Loss graphs. There are two methods for determining loss. The preferred method is to send 100 pings, and see how many fail to get back. There are two variants of the ping tests. The preferred method is to send the 100 pings concurrently with the throughput test (“concurrent pings”). In this case, the packet loss shown is for a period of significant network load. However, note that the throughput tests last 30 seconds. To send 100 pings in 30 seconds requires a ping spacing of 0.3 seconds. Many systems

do not permit this close spacing (1 second is generally the default minimum) without intervention of the sysadmin. On those systems where the sysadmin has chosen not to enable this spacing, the loss is determined from 100 pings prior to the throughput test ("standalone ping"). This thus shows the loss for a network with unknown load.

If pings are blocked, either at the source or destination, an alternative method is used that looks at the difference in the number of TCP packets retransmitted by the source node before and after the throughput test (obtained by `netstat -s`). It then calculates the percentage this represents of the total packets sent during the throughput test. Note that this alternative method has many drawbacks, and the results are subject to various errors. However, the thinking is that the data has been collected; it might mean something. For example, step changes probably do indicate that something has changed.

The first problem with this alternative method is that it measures not packet loss, but packet retransmission. While these are certainly related, a single packet lost may result in the retransmission of all following packets that had already been sent (unless sack is in use). So there is an unknown multiplier function applied due to this factor.

The next problem is that the retransmissions counted this way include all retransmissions for the source node, not just for the tcp session(s) to the destination node. So if the source node is retransmitting many packets to unrelated destinations, this count will be applied to the loss calculated here. This factor makes it essential not to attribute individual cases of high loss to the network between the source and destination tested here. In other cases, where the source node is sending out a high volume of data to many nodes (example: the GSFC GES DAAC at this writing is sending out about 250 Mbps, more or less continuously), a relatively small percentage of retransmission to unrelated nodes can overwhelm a low-rate flow to a test node.

The third problem is that the number of good packets is estimated, not calculated. The throughput rate and time is known from the throughput test, so the total data flow is known accurately. But the number of packets this represents is estimated using the maximum transmission unit (MTU), as reported by *iperf*. This could be correct, or there could have been more good packets sent – MTU is a maximum size, not guaranteed to be the size of every packet. Anyway, the loss rate is then calculated as the packets retransmitted divided by the sum of good packets and retransmitted packets. This third factor is probably not as big a source of error as the other factors.

For all the above reasons, this calculation of the percentage of retransmitted packets is not highly reliable.

RTT graphs. There are two methods used for determining RTT. The preferred method is to use the median value from the 100 pings. Even if the concurrent pings are used for the loss graph, the RTT is derived from standalone pings. This is because it has been found that in network-limited cases, if using concurrent pings, the pings will get queued at the bottleneck node behind packets from the throughput test. This will distort

the RTT value observed. If the pings are blocked, but the traceroute does indeed reach the destination node (yes, there are a few of these cases), the lowest of the three RTTs from the traceroute is used.

3.5 Troubleshooting Using Active-Measurements Results

One intended use of this system is automated event notification. It has been observed that network operation centers are quite good at detecting failure conditions, and responding appropriately. But they seem less sensitive to network degradation; it will take longer to recognize a higher packet-loss rate, or circuit-speed reduction, as long as some capability remains. It is hoped that the integrated measurements (see Section 4), when completed, will offer improved recognition and notification of these partial failure conditions. For this function, the recent performance from a source to a destination is compared to the expected or required value, and a status ("Good", "Adequate", "Poor", etc.) is assigned. If this status changes, automated email notifications are sent to appropriate parties.

Another use of this system is to assist troubleshooting of the problems observed. One characteristic of these tests, which contributes to the ability to troubleshoot problems, is the overlapping of test sources and sinks. Most sources test to several sinks, and most sinks are tested from at least two sources. So if most or all tests from a given source experience a performance change at about the same time, attention is directed near that source. Conversely, if performance changes are observed from multiple sources to a single sink, the change is attributed to the vicinity of the sink. Sometimes there are multiple sinks in relative proximity. This can allow estimation of the location of the problem. If both sinks are similarly affected, the change is inferred to be in a common element, while if only one is affected, the change is more local to that sink.

We conclude this section with some examples to show how inferences can be derived from actual graphs produced by the ENSIGHT system.

Example 1. This example involves diurnal variation. Examination of the "hourly" graphs of throughput (Figure 23) and packet loss (Figure 24) shows that high packet loss corresponds to low throughput at certain hours of the day, and that all sources are similarly affected. The inference is that there is a highly congested circuit along the route during U.S. East Coast business hours.

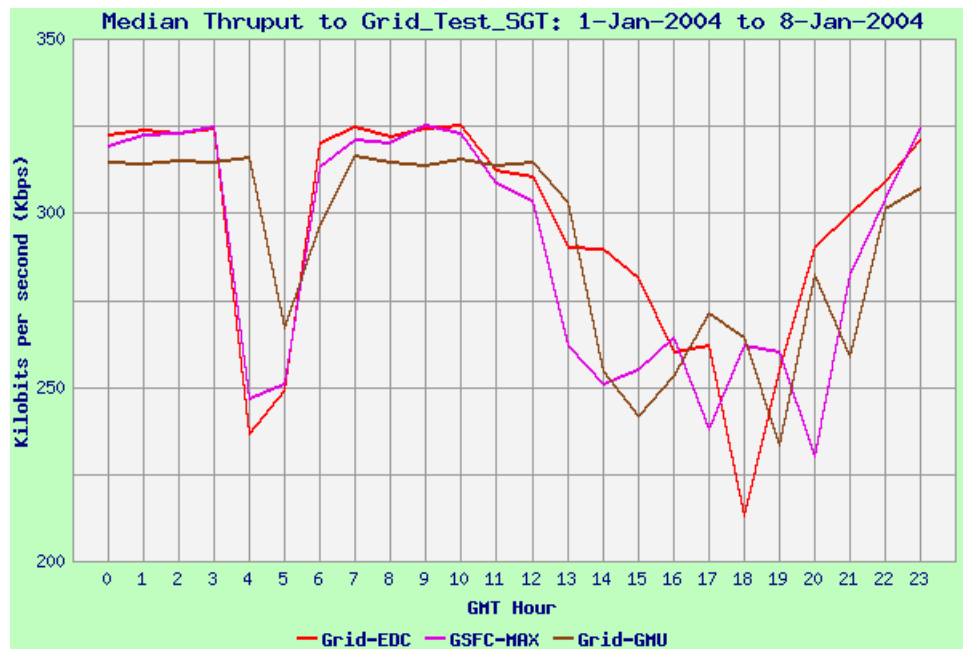


Figure 23. Throughput by GMT Hour

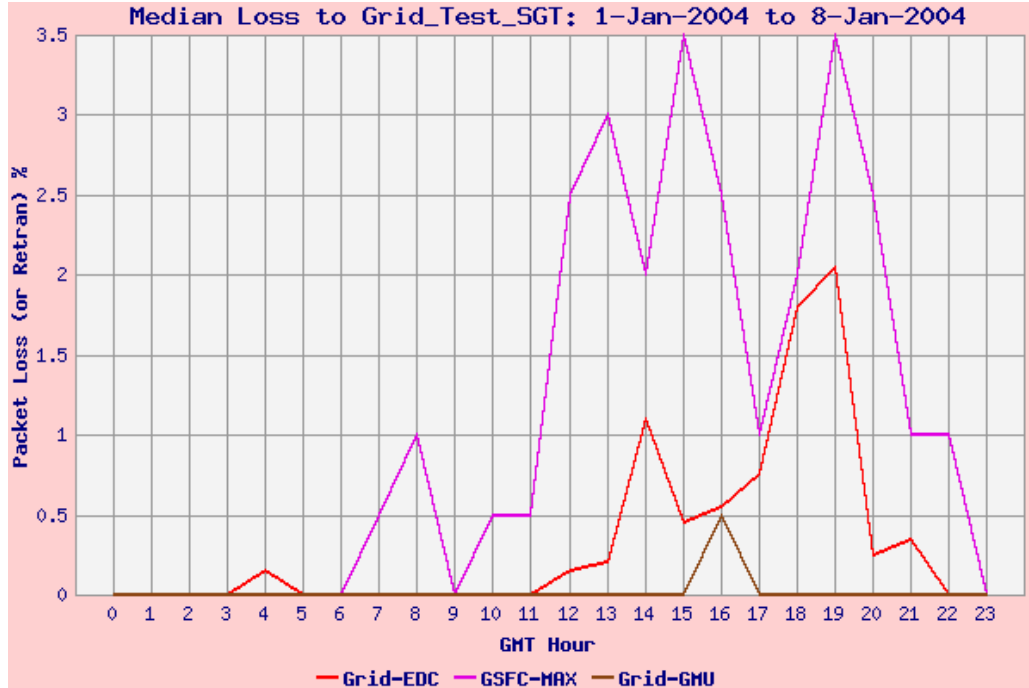


Figure 24. Packet Loss by GMT Hour

Example 2. This example involves a destination problem. In Figure 25 we see a noisy RTT develop from all four sources to Boston University (BU) on 30-September-2003. As the four sources are spread widely around the U.S., the inference is that the problem is near BU. Comparison of these results with another node near BU (perhaps the Massachusetts Institute of Technology (MIT)) would enable further pinpointing of the problem.

Example 3. The last example involves a source problem. In Figure 26 we see a high packet loss from one source to BU, while the other sources have low packet loss. The inference is that the problem is near the specific source.

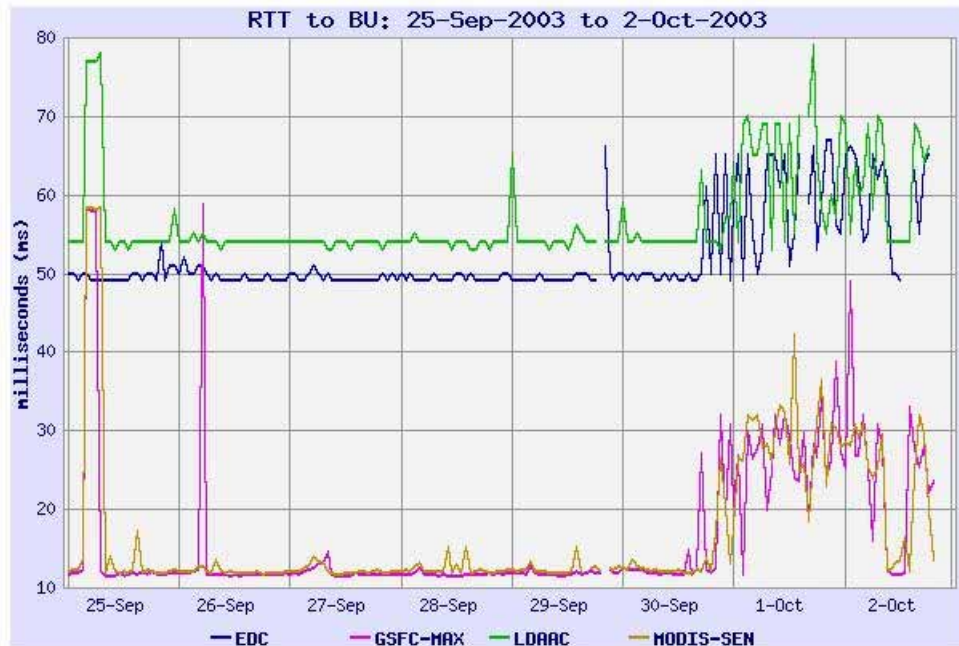


Figure 25. Round Trip Time

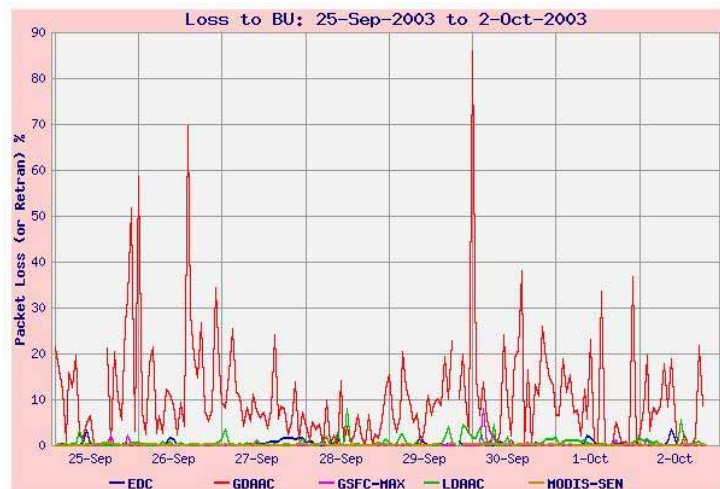


Figure 26. Packet Loss

4. Integrated Displays

In those cases where private circuits are used, the circuits are purchased to meet the requirements. One main question asked of this performance-measurement system is: Are these circuits performing up to their specifications. (This question cannot usually be asked about shared networks, because we typically have no information regarding their specifications, or the usage of these circuits by others. In these cases, the performance-measurement system focuses on whether the EOS requirements are being met).

It is clear that neither active nor passive measurements, by themselves, are sufficient to evaluate the circuits. The passive measurements only show flow if there is user demand. So if there is no user demand for some period (perhaps due to a problem with the user equipment or software), the passive measurements will show low utilization. But the circuit may very well be fine. The active measurements, conversely, compete for bandwidth with the user flows. So during periods of high user flow, the active measurements will be lower. The solution is to combine the active and passive measurements. The intent is to add the active and passive flows for the same period. In principle, the sum should indicate the total capacity of the network at the time. Figure 22 shows a typical integrated graph.

There are several factors and difficulties in combining the active and passive results that should be described. One problem is that ideally, the active and passive measurements should be taken over the same time periods. Currently, however, the active throughput tests run for 30 seconds. The user flows typically last much longer. If so, the flow rate is averaged over their duration.

So adjustments are made to the data before adding the passive to the active results. The first is due to recognition that the measurements are being made at different levels of the protocol stack. The active tests measure TCP payload – they are at the application protocol layer. But the user flows measure the length of the entire IP packet. So the passive measurements are "discounted" by 3% as an approximation for the layer 3 and 4 protocol overhead. Next, it is noted that the data flows of the active tests are included in the passive measurements. So this effect is subtracted out of the passive values.

Finally, an "interference effect" is estimated and adjusted for. In this case, if a user flow lasts longer than the *iperf* test, it will only compete for bandwidth with the *iperf* test for a short percentage of its duration. Thus its average throughput rate will be substantially unaffected by the *iperf* test. However, its rate DURING the *iperf* test WILL be affected

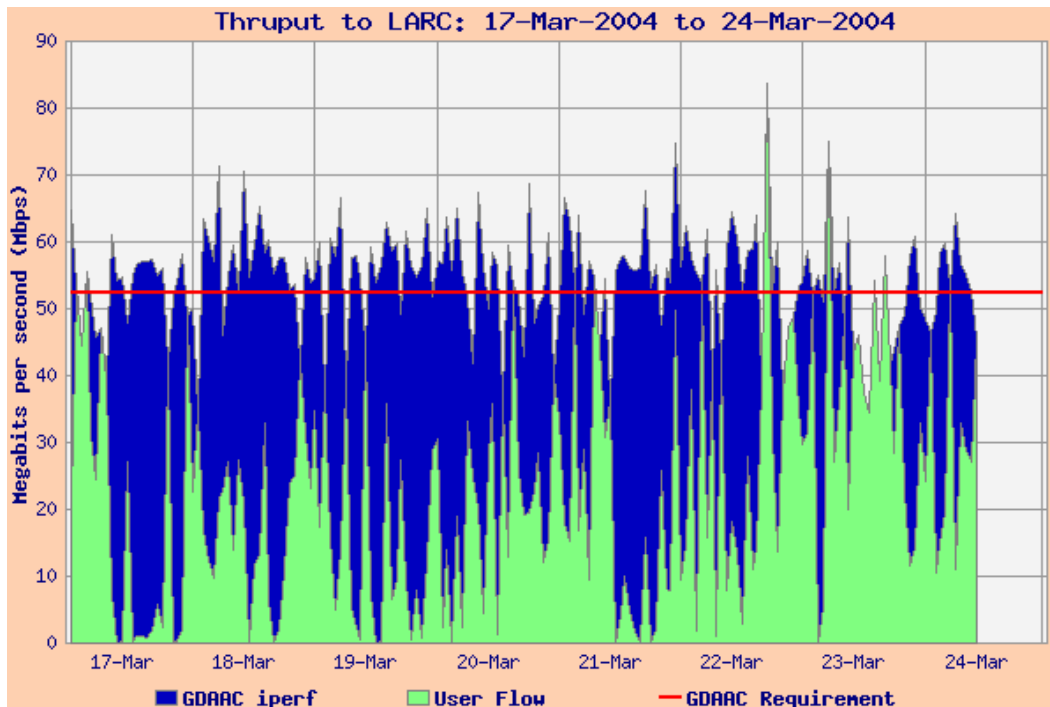


Figure 22. Sample Integrated Graph

by the *iperf* test, and thus be significantly lower. For example, if the user flow consumes the full bandwidth (100%) of a circuit for a long period, it may get only 50% during the *iperf* test, if both flows are a single stream – the *iperf* test would get the other 50% during its 30 second lifetime. The user flow's pro-rated flow rate would still be close to 100%, since it is averaged over the full duration of the flow. If the user flow's 100% was added to *iperf*'s 50%, the total would be 150% of the circuit bandwidth – an unrealistic value. So an attempt is made to estimate the interference of the *iperf* test on the user flow. This adjustment is made to the *iperf* value, to allow the user-flow portion of the graph to accurately reflect the observed flow.

Although it is recognized that this method has inherent inaccuracies, it is hoped that it is still useful. The example graph in Figure 22 above shows a much flatter total result (as expected on a private circuit) than either the user flow or *iperf* would. It is clear that the *iperf* values drop when the user flow increases, and vice versa.

5. Conclusion

In this paper we have described features of the web-based ENSIGHT network-monitoring tool. EOSDIS personnel use ENSIGHT to detect and troubleshoot performance problems, track utilization of network resources, verify requirements against performance data, and forecast required upgrades. Operators of individual university networks and EOSDIS partner networks, such as Abilene, also use ENSIGHT to help them diagnose problems on their own networks.

Through a combination of active, passive, and integrated measurements, our network-monitoring system provides a comprehensive view of relevant performance parameters. The ENSIGHT web site provides several options for visualizing the performance data that is collected. This flexibility enables users to view data in the context that is most meaningful to them, e.g., in the context of a specific mission. In this paper we have provided insight into how the web site can be used and the information that can be gleaned from the statistics that are collected.

Finally, we are continually striving to improve the ENSIGHT tool by incorporating new features, new ways of presenting the data, and new techniques for collecting the data.

References

- [1] The Earth Observing System Project Science Office, <http://eospsso.gsfc.nasa.gov>
- [2] Aura website, <http://aura.gsfc.nasa.gov/>
- [3] NASA Integrated Services Network, <http://www.nisn.nasa.gov>
- [4] Abilene Backbone Network, <http://abilene.internet2.edu/>
- [5] ENSIGHT Performance Measurements, <http://ensight.eos.nasa.gov>
- [6] Net-SNMP Project Home Page, <http://net-snmp.sourceforge.net/>
- [7] GD.pm - Interface to GD Graphics Library, <http://stein.cshl.org/WWW/software/GD/>
- [8] Perl DBI, <http://dbi.perl.org/>
- [9] Flow-tools, <http://www.splintered.net/sw/flow-tools>
- [10] Apache http Server Project, <http://httpd.apache.org>
- [11] Cisco White Paper: NetFlow Services and Applications, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm
- [12] MRTG: The Multi Router Traffic Grapher, <http://mrtg.hdl.com/mrtg.html>
- [13] RRDtool – Round Robin Database Tool, <http://www.caida.org/tools/utilities/rrdtool/>
- [14] Iperf Version 1.7.0, <http://dast.nlanr.net/Projects/Iperf/>
- [15] NLANR, <http://www.nlanr.net/>
- [16] ttcp/nttcp/nuttcp/iperf versions, <http://sd.wareonearth.com/~phil/net/ttcp>

- [17] TCPWatch, <http://hathaway.freezope.org/Software/TCPWatch>
- [18] Committee on Earth Observation Satellites home page, <http://www.ceos.org>
- [19] ESTO home page, <http://esto.nasa.gov>
- [20] NASA's Computational Technologies Project home page,
<http://sdcd.gsfc.nasa.gov/ESS/>